The status of claims in the case is as follows:

1 1. [Original] A method for control and management of

2 communication traffic, comprising the steps of:

3   expressing access rules as filters referencing system

4   kernel data;

5   for outbound processing, determining source application

6   indicia;

7   for inbound packet processing, executing a look-ahead

8   function to determine target application indicia; and

9   responsive to said source or target application

10   indicia, executing filter processing.

1 2. [Original] The method of claim 1, further comprising

2 the steps of executing said determining and executing steps

3 within a kernel filtering function upon encountering a

4   filter selector field referencing kernel data not included

5   in said packet.

1   3.   [Original] The method of claim 1, said filter

2   processing including the steps of:

3   determining a task or thread identifier;

4   based on said task or thread identifier, determining a

5   process or job identifier; and

6   based on said process or job identifier, determining

7   job or process attributes for filter processing.

1   4.   [Original] The method of claim 1, said filter

2   processing including the steps of:

3   determining a user identifier; and

4   based on said user identifier, determining user

5   attributes for filter processing.

1   5.   [Original]   The method of claim 3, further comprising

2   the step of determining from said task identifier a work

3   control block containing said process or job identifier.

1   6.   [Original]   The method of claim 1, further comprising

2   the steps for inbound processing of:

3   passing an inbound packet to a sockets layer to

4   identify said target application.

1   7.   [Original]   The method of claim 6, further comprising

2   the step of marking said inbound packet as not deliverable

3   before passing it to said sockets layer.

1   8.   [Original]   The method of claim 1, further comprising

2   the steps of:

3   delivering to said filters infrastructure access rules

4   for defining security context.

1   9.   [Original]   The method of claim 8, said infrastructure

2   including logging, auditing, and filter rule load controls.

1   10.   [Original]   A method for control and management of

2   aspects of communication traffic within filtering,

3   comprising the steps of:

4   receiving IP packet data into a TCP/IP protocol stack

5   executing within a system kernel

6   executing filtering code within said system kernel with

7   respect to non-IP packet data accessed within said

8   system kernel outside of said TCP/IP protocol stack.

1   11.   [Original]   The method of claim 10, said non-IP packet

2   data including context data regarding said IP packet.

1   12.   [Original]   The method of claim 10, said non-IP packet

2   data including data specific to a task generating said non-

3   IP packet data.

1   13.   [Original]   The method of claim 10, said non-IP packet

2   data including data specific to a task that will receive

3   said IP packet.

1   14.   [Original]   The method of claim 11, said context data

2   including packet arrival interface indicia.

1   15.   [Original]   The method of claim 11, said context data

2   including packet arrival time-of-day indicia.

1   16.   [Original]   The method of claim 10, further comprising

2   the steps of:

3   establishing a tunnel between two IP address limiting

4   traffic to applications bound to ports at each end of

5   said tunnel;

6   said filtering code accessing filtering attributes

7   further limiting traffic selectively to job indicia.

1   17. [Original]   The method of claim 10, further comprising

2   the steps of:

3   establishing a tunnel between two IP address limiting

4   traffic to applications bound to ports at each end of

5   said tunnel; and

6   said filtering code accessing filtering attributes

7   further limiting traffic selectively to user

8   identification indicia.

1   18. [Original]   A method for centralizing system-wide

2   communication management and control within filter rules,

3   comprising the steps of:

4   providing filter statements syntax for accepting

5   parameters in the form of a selector, each selector

6   specifying selector field, operator, and a set of

7   values; and

8   said selector referencing data that does not exist in

9   IP packets.

1   19. [Original] The method of claim 18, said parameters

2   selectively including userid, user profile, user class, user

3   group, user group authority, user special authority, job

4   name, process name, job group, job class, job priority,

5   other job or process attributes, and date & time.

1   20. [Original] The method of claim 18, said filters

2   statements being provided within a user interface to said

3   system.

1   21. [Original] The method of claim 18, further comprising

2   the steps of:

3   establishing a tunnel between two IP address limiting

4   traffic to applications bound to ports at each end of

5   said tunnel;

6   said filtering code accessing filtering attributes

7   further limiting traffic selectively to job indicia;

8   and

9   operating said filtering code within a kernel filtering

10   function upon encountering a filter selector field

11   referencing kernel data not included in said traffic.

1   22.   [Original]   A method for traversing a portion only of a

2   protocol stack to disallow selective IP packet traffic,

3   comprising the steps of:

4   receiving a packet in the kernel of the operating

5   system of a first node from an application, said kernel

6   including a filter processor;

7   for inbound packet processing to a first node from a

8   second node, executing a look-ahead function in the

9   system kernel of said first node to determining a

10   target application;

11   for both said inbound packet processing, and for

12   outbound packet processing from said first node to said

13   second node, executing within said kernel the steps of

14   processing said packet by determining a task ID;

15   responsive to said task ID, determining a

16   corresponding work control block;

17   determining a user ID, process or job identifier
18   from said work control block;

19   from the user ID, process or job identifier
20   selectively determining attributes for said user
21   process or job; and

22   passing said attributes to said filter processor
23   for managing and controlling communication
24   traffic.

1   23.   [Original]   A method for expressing access rules as
2   filters, comprising the steps of:

3   providing a filter statements syntax for accepting
4   parameters in the form of a selector, each selector
5   specifying selector field, operator, and a set of
6   values; and

7   said selector referencing data that does not exist in
8   IP packets for controlling access to an application.

1   24.   [Original]   A method for managing and controlling

2   communication traffic by centralizing access rules in

3   filters executing within and referencing data available in

4   system kernels, comprising the steps for outbound packet

5   processing from a first node to a second node of:

6   receiving said packet in the kernel of the operating

7   system of said first node from an application or

8   process at said first node;

9   processing said packet by determining a task ID;

10   responsive to said task ID, determining a corresponding

11   work control block;

12   responsive to said work control block, determining a

13   process or job identifier;

14   responsive to said process or job identifier,

15   determining job or process attributes.

1   25. [Original] The method of claim 24, further comprising

2   the steps for inbound packet processing from said second

3   node to said first node of:

4   initially operating said kernel at said first node to

5   determine a target application for said packet at said

6   first node.

1   26. [Original] The method of claim 25, said initially

2   operating step comprising executing a look-ahead function.

1   27. [Original] The method of claim 26, said look-ahead

2   function including the steps of operating a filter function

3   to request of a sockets layer the identity of an application

4   to which said sockets layer would pass said packet.

1   28. [Original] The method of claim 27, further comprising

2   the step of marking said packet as non-deliverable and

3   thereafter passing said packet to said sockets layer to

4   identify said application.

1   29.   [Original]   A method for managing and controlling

2   communication traffic by centralizing the access rules,

3   comprising the steps for outbound packet processing from a

4   first node to a second node of:

5   receiving said packet in the kernel of the operating

6   system of said first node from an application or

7   process at said first node, said kernel including a

8   filter processor;

9   processing said packet by determining a task ID;

10   responsive to said task ID, determining a corresponding

11   work control block;

12   determining a user ID control block from said work

13   control block;

14   from the user ID control block determining attributes

15   for said user; and

16   passing said attributes to said filter processor for

17   managing and controlling communication traffic.

1   30.   [Original]   The method of claim 29, further comprising

2   the steps for inbound packet processing from said second

3   node to said first node of:

4   initially operating said kernel at said first node to

5   determine a target application for said packet at said

6   first node.

1   31.   [Original]   The method of claim 30, said initially

2   operating step comprising executing a look-ahead function.

1   32.   [Original]   The method of claim 31, said look-ahead

2   function including the steps of operating a filter function

3   to request of a sockets layer the identity of an application

4   to which said sockets layer would pass said packet.

1   33.   [Original]   The method of claim 32, further comprising

2   the step of marking said packet as non-deliverable and

3   thereafter passing said packet to said sockets layer to

4   identify said application.

1   34.   [Original]   A method for control and management of

2   communication traffic with respect to a system node,

3   comprising the steps of:

4   receiving at said system node an inbound packet; and

5   executing within a protocol stack of the system kernel

6   of said system node a filtering function identifying

7   for said inbound packet a filter referencing non-packet

8   data; and

9   responsive to said filter, executing a look-ahead

10   function for identifying a target application for said

11   inbound packet.

1   35.   [Original]   The look-ahead function of the method of

2   claim 34 further comprising the steps of:

3   passing to a transport layer function identified by an

4   IP header a packet marked non-deliverable for

5   determining which user-level process or job is to

6   receive said packet;

```
7          receiving from said transport layer an application

8          layer task identifier for said user-level process or

9          job; and thereafter


10         passing said packet marked by said task identifier to

11         said transport layer for delivery to said application

12         layer task.



1     36.  [Original]  System for control and management of

2     communication traffic, comprising:


3          a system kernel including a filter function and stack

4          data;


5          said filter function including a filter selectively

6          referencing said stack  data for expressing access

7          rules;


8          said filter function being responsive to receipt of an

9          outbound packet for determining a source application;


10         said filter function being responsive to receipt of an

11         inbound packet processing for executing a look-ahead
```

12   function to determine a target application; and

13   said filter function being responsive to said source or
14   target application for executing filter processing.

1   37. [Original] A system for control and management of
2   aspects of communication traffic within filtering,
3   comprising:

4   a system kernel;

5   a protocol stack executing within said system kernel
6   for receiving IP packet data; and

7   filtering code within said system kernel operable with
8   respect to non-IP packet data accessed within said
9   system kernel outside of said protocol stack for
10   controlling and managing said aspects of communication
11   traffic.

1   38. [Original] A system for centralizing system-wide
2   communication management and control within filter rules,

3   comprising:

4   filter statements having a syntax for accepting

5   parameters in the form of a selector, each selector

6   specifying selector field, operator, and a set of

7   values; and

8   said selector referencing data that does not exist in

9   IP packets.

1   39.   [Original]   A system for traversing a portion only of a

2   protocol stack to disallow selective IP packet traffic,

3   comprising:

4   a system kernel;

5   a filter processor executing within said system kernel;

6   said filter processor responsive to an inbound packet

7   for executing a look-ahead function for determining a

8   target application;

9   said filter processor responsive to both inbound and

10   outbound packets for

11   processing said packet by determining a task ID;

12   responsive to said task ID, determining a
13   corresponding work control block;

14   determining a user ID, process or job identifier
15   from said work control block;

16   from the user ID, process or job identifier
17   selectively determining attributes for said user
18   process or job; and

19   passing said attributes to said filter processor
20   for managing and controlling communication
21   traffic.

1   40.   [Original]   A system for expressing access rules as
2   filters, comprising:

3   a filter statements for accepting parameters in the
4   form of a selector, each selector specifying selector

5   field, operator, and a set of values; and

6   said selector referencing data that does not exist in

7   IP packets for controlling access to an application.

1   41. [Original] A system for managing and controlling

2 communication traffic by centralizing access rules in

3 filters executing within and referencing data available in

4 system kernels, comprising:

5   code for receiving a packet in the kernel of the

6   operating system of a first node from an application or

7   process at said first node;

8   code for processing said packet by determining a task

9   ID;

10   code responsive to said task ID for determining a

11   corresponding work control block;

12   code responsive to said work control block for

13   determining a process or job identifier; and

14   code responsive to said process or job identifier for

15   determining job or process attributes.

1   42. [Original] A system for managing and controlling

2 communication traffic by centralizing access rules,

3 comprising:

4   a first system node;

5   a second system node;

6   a kernel of the operating system of said first system

7   node including a kernel filter processor;

8   said kernel for receiving from an application or

9   process at said first system node a packet for

10   communication to said second system node;

11   said kernel further for processing said packet by

12   determining a task ID; responsive to said task ID,

13   determining a corresponding work control block;

14   determining a user ID control block from said work

15   control block; from the user ID control block

16   determining attributes for said user; and passing said

17   attributes to said system kernel filter processor for

18   managing and controlling communication traffic.

1   43. [Original] A system for control and management of

2   communication traffic with respect to a system node,

3   comprising:

4   a filtering function executing within a protocol stack

5   of the system kernel of said system node identifying

6   for an inbound packet a filter referencing non-packet

7   data; and

8   a look-ahead function responsive to said filter for

9   identifying a target application for said inbound

10   packet.

1   44. [Original] A program storage device readable by a

2   machine, tangibly embodying a program of instructions

3   executable by a machine to perform method steps for control

4   and management of communication traffic, said method steps

5   comprising:

6   expressing access rules as filters referencing system

7   kernel data;

8   for outbound processing, determining a source

9   application;

10   for inbound packet processing, executing a look-ahead

11   function to determine a target application; and

12   responsive to said source or target application,

13   executing filter processing.

1   45.   [Currently amended] A ~~program storage device readable~~

2 ~~by a machine, tangibly embodying a program of instructions~~

3 ~~executable by a machine to perform method steps~~ computer

4 program product for control and management of aspects of

5 communication traffic within filtering, said ~~method steps~~

6 computer program product comprising:

7   a computer readable medium;

8   ~~receiving~~ first program instructions to receive IP

9   packet data into a TCP/IP protocol stack executing

10   within a system kernel; and

11   ~~executing~~ second program instructions to execute

12   filtering code within said system kernel with respect

13   to non-IP packet data accessed within said system

14   kernel outside of said TCP/IP protocol stack; and

15   wherein

16   said first and second program instructions are recorded

17   on said medium.

1   46. [Currently amended] A ~~program storage device readable~~

2   ~~by a machine, tangibly embodying a program of instructions~~

3   ~~executable by a machine to perform method steps~~ a computer

4   program product for centralizing system-wide communication

5   management and control within filter rules, said ~~method~~

6   ~~steps~~ computer program product comprising:

7   a computer readable medium;

8   ~~providing~~ first program instructions to filter

9   statements syntax for accepting parameters in the form

10   of a selector, each selector specifying selector field,

11   operator, and a set of values; and

12   second program instructions to cause said selector

13   ~~referencing~~ to reference data that does not exist in IP

14   packets; and wherein

15   said first and second program instructions are recorded

16   on said medium.

1   47.   [Currently amended]   A ~~program storage device readable~~

2   ~~by a machine, tangibly embodying a program of instructions~~

3   ~~executable by a machine to perform method steps~~ a computer

4   program product for managing and controlling communication

5   traffic by centralizing access rules in filters executing

6   within and referencing data available in system kernels,

7   said ~~method steps~~ computer program product comprising:

8   a computer readable medium;

9   ~~receiving~~ first program instructions to receive said

10   packet in the kernel of the operating system of said

11   first node from ~~an application or~~ a process at said

12   first node;

13   ~~processing~~ <u>second program instructions to process</u> said

14   packet by determining a task ID;

15   <u>third program instructions,</u> responsive to said task ID,

16   ~~determining~~ <u>to determine</u> a corresponding work control

17   block;

18   <u>fourth program instructions,</u> responsive to said work

19   control block, ~~determining~~ <u>to determine</u> a process or

20   job identifier; <u>and</u>

21   <u>fifth program instructions,</u> responsive to said process

22   or job identifier, ~~determining~~ <u>to determine</u> job or

23   process attributes<u>; and wherein</u>

24   <u>said first, second, third, fourth, and fifth program</u>

25   <u>instructions are recorded on said medium</u>.

1   48.   [Currently amended]   The <u>computer</u> program ~~storage~~

2   ~~device~~ <u>product</u> of claim 47, said ~~method steps~~ <u>computer</u>

3   <u>program product</u> further comprising for inbound packet

4   processing from said second node to said first node:

5   sixth program instructions to initially ~~operating~~

6   operate said kernel at said first node to determine a

7   target application for said packet at said first node;

8   and wherein

9   said sixth program instructions are recorded on said

10   medium.

1   49. [Original] A computer program product or computer

2 program element for control and management of communication

3 traffic according to the steps comprising:

4   expressing access rules as filters referencing system

5   kernel data;

6   for outbound processing, determining a source

7   application;

8   for inbound packet processing, executing a look-ahead

9   function to determine a target application; and

10   responsive to said source or target application,

11   executing filter processing.

1   50.   [Original]   A computer program product or computer

2   program element for control and management of aspects of

3   communication traffic within filtering according to steps

4   comprising:

5   receiving IP packet data into a TCP/IP protocol stack

6   executing within a system kernel

7   executing filtering code within said system kernel with

8   respect to non-IP packet data accessed within said

9   system kernel outside of said TCP/IP protocol stack.

1   51.   [Original]   A computer program product or computer

2   program element for centralizing system-wide communication

3   management and control within filter rules according to

4   method steps comprising:

5   providing filter statements syntax for accepting

6   parameters in the form of a selector, each selector

7   specifying selector field, operator, and a set of

8   values; and

9 said selector referencing data that does not exist in

10 IP packets.

1 52. [Original] A computer program product or computer

2 program element for managing and controlling communication

3 traffic by centralizing access rules in filters executing

4 within and referencing data available in system kernels

5 according to method steps comprising:

6 receiving said packet in the kernel of the operating

7 system of said first node from an application or

8 process at said first node;

9 processing said packet by determining a task ID;

10 responsive to said task ID, determining a corresponding

11 work control block;

12 responsive to said work control block, determining a

13 process or job identifier;

14 responsive to said process or job identifier,

15 determining job or process attributes.

1   53.   [Original]   The computer program product or element of

2   claim 52, said method steps further comprising for inbound

3   packet processing from said second node to said first node:

4   initially operating said kernel at said first node to

5   determine a target application for said packet at said

6   first node.

~